OPERATING SYSTEM

Dr.Eng. Ahmed Moustafa Elmahalawy

Computer Science and Engineering

ميثاق المحاضرة

الإلتزام بالوقت

المشاركة

تحديد الهدف

إغلاق المحمول

الأحترام المتبادل

Course out line:-

I- Introduction to operating systems.

II - Process Management.

III Storage Management.

IV- File Management.

IIV- I/O Management.

IIIV- Networking, protection and security.

Chapter 1
Introduction
to
operating systems

Contents

1.1 What Operating Systems Do?

1.2 Computer-System Organization

1.3 Computer-System Architecture

1.4 Operating-System Structure

1.5 Operating-System Operations

1.6 Special-Purpose Systems

1.7 Computing Environments

An **operating system** is a program that manages the computer hardware.

It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.

Operating systems for handheld computers are designed to provide an environment in which a user can easily interface with the computer to execute programs.

Thus, some **operating systems** are designed to be *convenient,* others to be *efficient,* and others some combination of the two.

1.1 What Operating Systems Do

A computer system can be divided roughly into four components: *(Figure 1)*

- the *hardware,*
- the *operating system,*
- the *application programs,*
- the *users.*

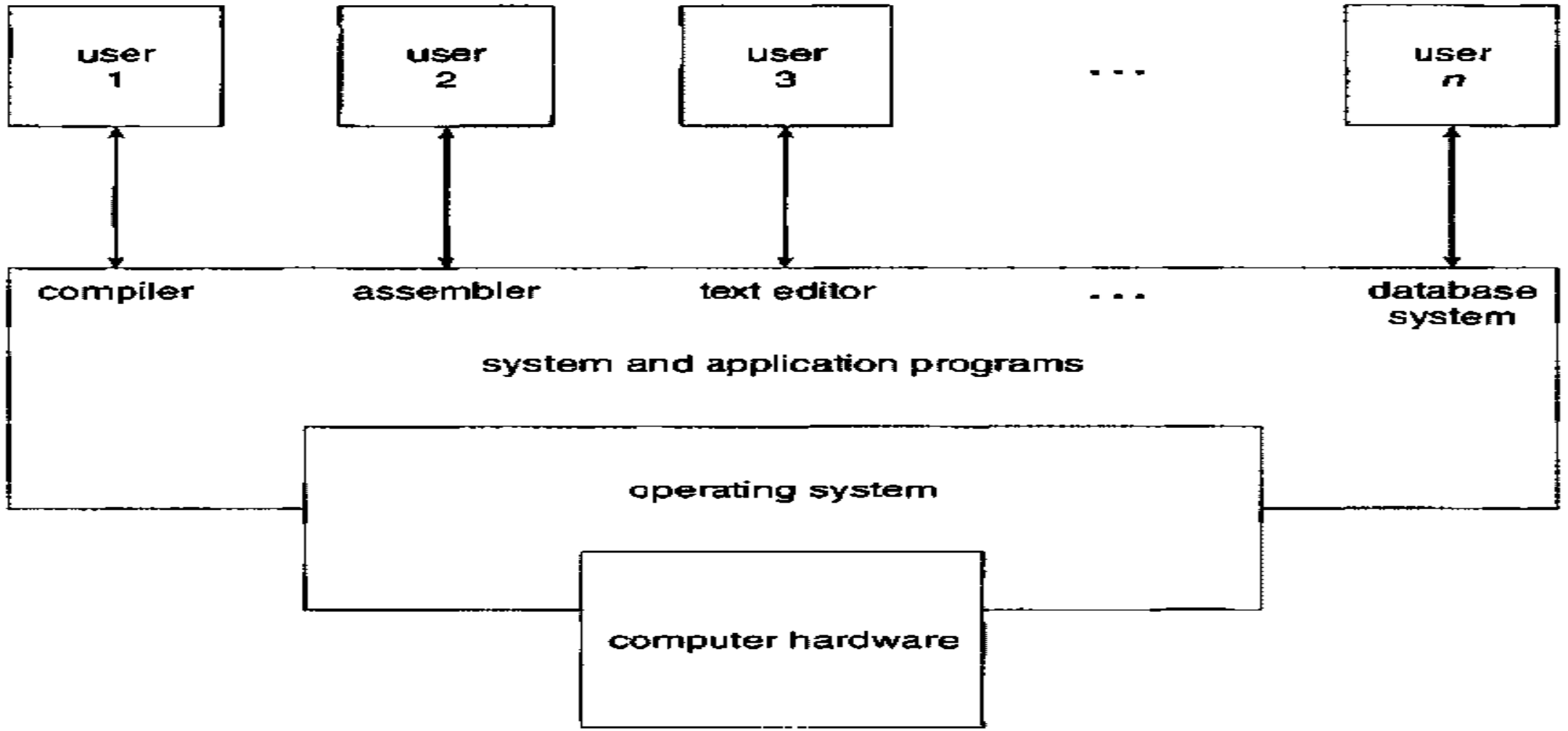

Figure 1 Abstract view of the components of a computer system.

The operating system controls and coordinates the use of the hardware among the various application programs for the various users.

The operating system provides the means for proper use of these resources in the operation of the computer system.

From the user's point of view, The operating system is designed mostly for ease of **use,** with some attention paid to performance and none paid to **resource utilization**—how various hardware and software resources are shared.

Performance is, of course, important to the user; but rather than resource utilization, such systems are optimized for the single-user experience.

From the computer's point of view, the operating system is the program most intimately involved with the hardware.

In this context, we can view an operating system as a **resource allocator.**

A computer system has many resources that may be required to solve a problem.

The operating system acts as the manager of these resources efficiently and fairly.

How, though, can we define what an operating system is? In general, we have no completely adequate definition of an operating system.

Operating systems exist because they offer a reasonable way to solve the problem of creating a usable computing system.

The fundamental goal of computer systems is to execute user programs and to make solving user problems easier.

The common functions of controlling and allocating resources are then brought together into one piece of software: the operating system.

A more common definition is that the operating system is the one program running at all times on the computer (usually called the **kernel),** with all else being systems programs and application programs.

1.2 Computer-System Organization

A modern general-purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access to shared memory (Figure 2).

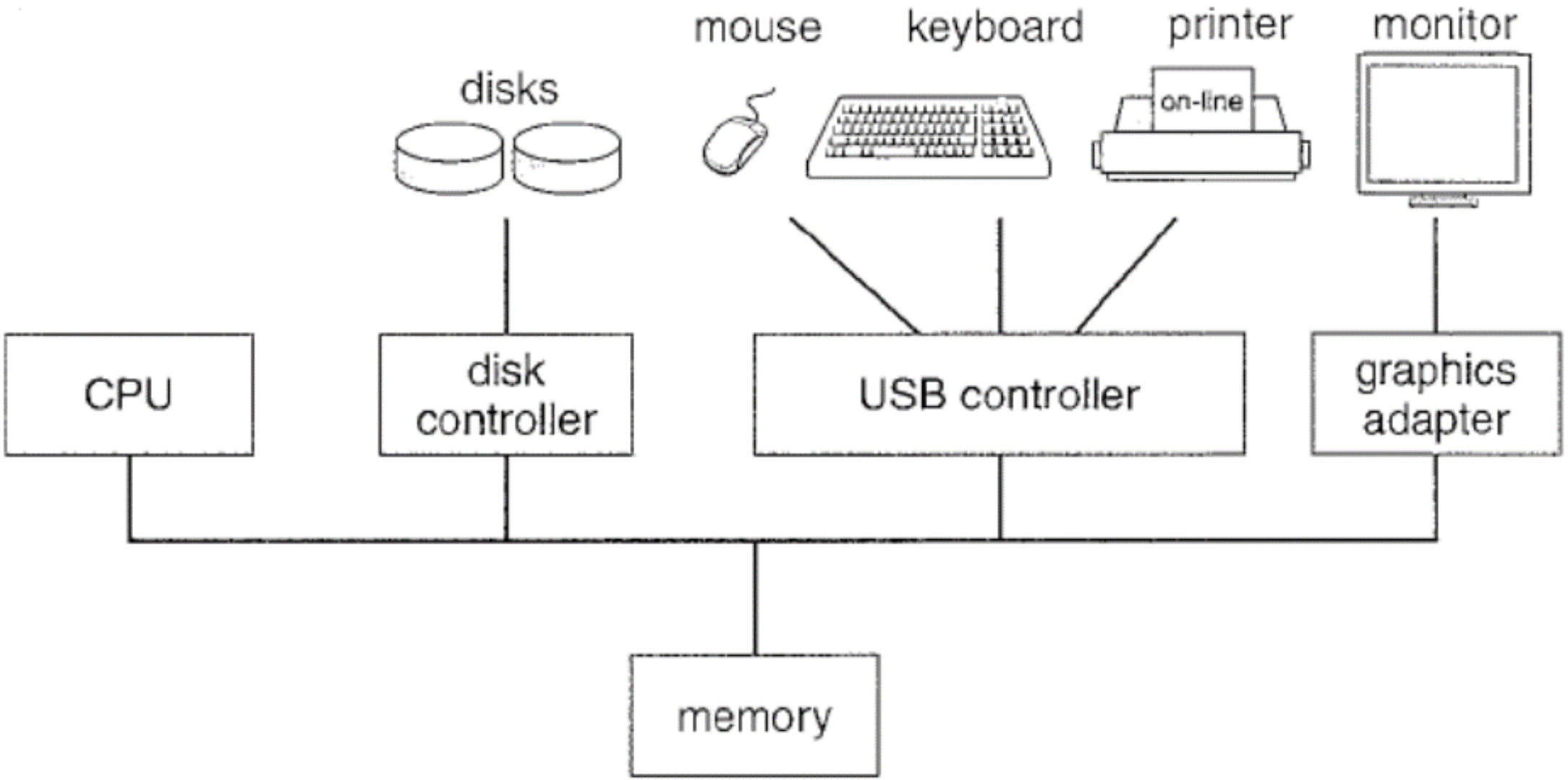

Figure 2 A modern computer system.

1.3 Computer-System Architecture

A computer system may be organized in a number of different ways, which we can categorize roughly according to the number of general-purpose processors used.

1- Single-Processor Systems

2- Multiprocessor Systems

3- Clustered Systems

1.4 Operating-System Structure

An operating system provides the environment within which programs are executed.

Internally, operating systems vary greatly in their makeup, since they are organized along many different lines.

One of the most important aspects of operating systems is the ability to multiprogram.

A single user cannot, in general, keep either the CPU or the I/O devices busy at all times.

Multiprogramming increases CPU utilization by organizing jobs (code and data) so that the CPU always has one to execute.

The idea is as follows: The operating system keeps several jobs in memory simultaneously (Figure 3).

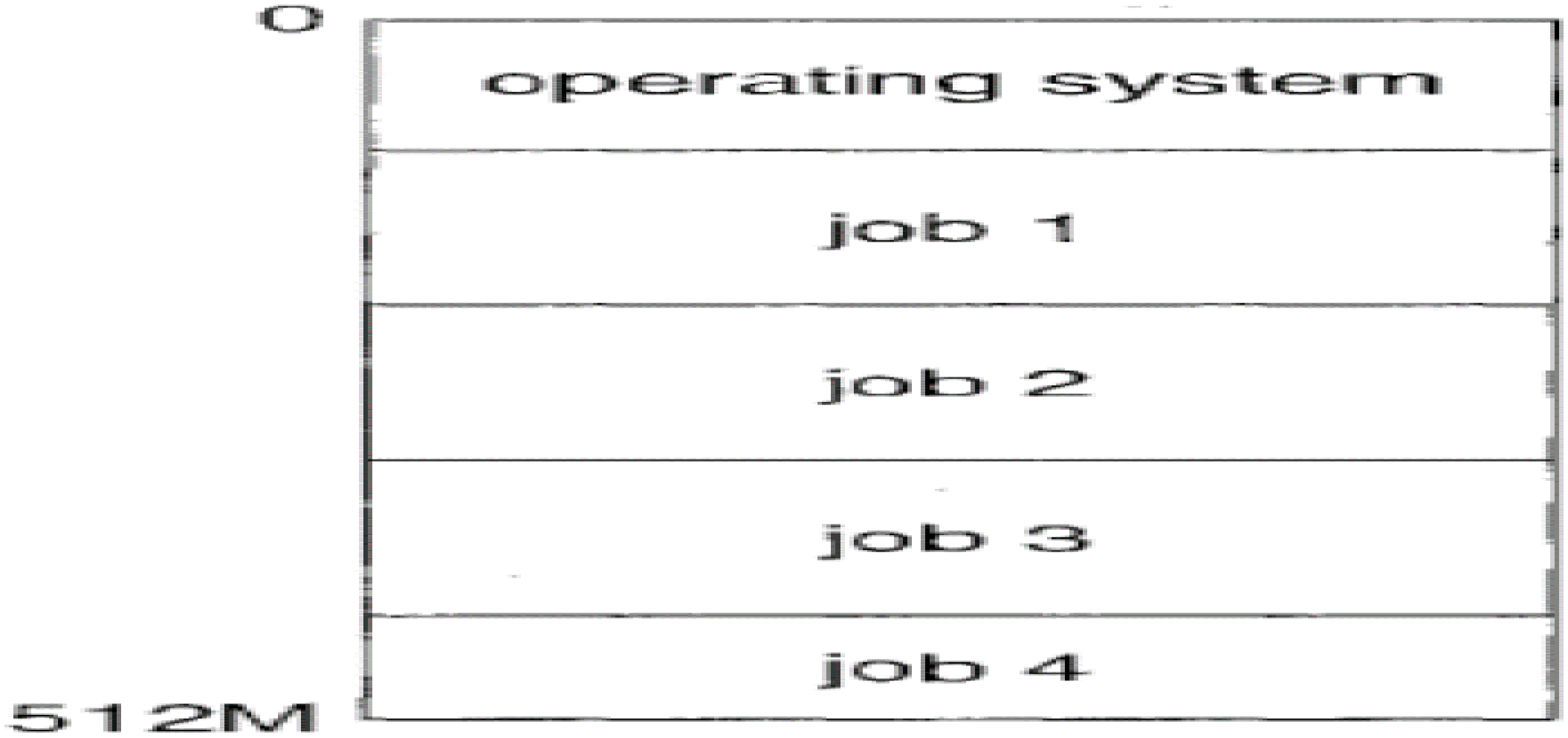

Figure 3 Memory layout for a multiprogramming system.

Time sharing (or **multitasking)** is a logical extension of multiprogramming.

In time-sharing systems, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.

Time sharing requires an **interactive** (or **hands-on) computer system,** which provides direct communication between the user and the system.

A time-shared operating system allows many users to share the computer simultaneously.

Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user.

A time-shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.

1.5 Operating-System Operations

Modern operating systems are **interrupt driven.** If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen.

The interrupt-driven nature of an operating system defines that system's general structure.

For each type of interrupt, separate segments of code in the operating system determine what action should be taken.

An interrupt service routine is provided that is responsible for dealing with the interrupt.

1. Dual-Mode Operation

In order to ensure the proper execution of the operating system, we must be able to distinguish between the execution of operating-system code and userdefined code.

The approach taken by most computer systems is to provide hardware support that allows us to differentiate among various modes of execution.

2. Timer

We must ensure that the operating system maintains control over the CPU.

We must prevent a user program from getting stuck in an infinite loop or not calling system services and never returning control to the operating system.

To accomplish this goal, we can use a **timer.** A timer can be set to interrupt the computer after a specified period. The period may be fixed or variable.

3. Process Management

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes.
- Suspending and resuming processes.
- Providing mechanisms for process synchronization.
- Providing mechanisms for process communication.
- Providing mechanisms for deadlock handling.

4. Memory Management

The operating system is responsible for the following activities in connection with memory management:

- Keeping track of which parts of memory are currently being used and by whom.
- Deciding which processes (or parts thereof) and data to move into and out of memory.
- Allocating and deallocating memory space as needed.

5. Storage Management

To make the computer system convenient for users, the operating system provides a uniform, logical view of information storage.

The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file.

The operating system maps files onto physical media and accesses these files via the storage devices.

5.1 File-System Management

File management is one of the most visible components of an operating system.

Computers can store information on several different types of physical media.

Magnetic disk, optical disk, and magnetic tape are the most common.

Each of these media has its own characteristics and physical organization.

The operating system is responsible for the following activities in connection with file management:

- Creating and deleting files
- Creating and deleting directories to organize files
- Supporting primitives for manipulating files and directories
- Mapping files onto secondary storage
- Backing up files on stable (nonvolatile) storage media

5.2 Mass-Storage Management

As we have already seen, because main memory is too small to accommodate all data and programs, and because the data that it holds are lost when power is lost, the computer system must provide secondary storage to back up main memory.

The operating system is responsible for the following activities in connection with disk management:

- Free-space management
- Storage allocation
- Disk scheduling

5.3 Caching

Caching is an important principle of computer systems. Information is normally kept in some storage system (such as main memory).

As it is used, it is copied into a faster storage system—the cache—on a temporary basis.

When we need a particular piece of information, we first check whether it is in the cache.

If it is, we use the information directly from the cache; if it is not, we use the information from the source, putting a copy in the cache under the Assumption.

5.4 I/O Systems

One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. For example, in UNIX, the peculiarities of I/O devices are hidden from the bulk of the operating system itself by the I/O **subsystem.**

The I/O subsystem consists of several components:

- A memory-management component that includes buffering, caching, and spooling.
- A general device-driver interface.
- Drivers for specific hardware devices.

5.5 Protection and Security

If a computer system has multiple users and allows the concurrent execution of multiple processes, then access to data must be regulated.

For that purpose, mechanisms ensure that files, memory segments, CPU, and other resources can be operated on by only those processes that have gained proper authorization from the operating system.

Protection and security require the system to be able to distinguish among all its users.

Most operating systems maintain a list of user names and associated **user identifiers (user IDs).**

1.6 Distributed Systems

A distributed system is a collection of physically separate, possibly heterogeneous computer systems that are networked to provide the users with access to the various resources that the system maintains.

Access to a shared resource increases computation speed, functionality, data availability, and reliability.

Some operating systems generalize network access as a form of file access, with the details of networking contained in the network interface's device driver. Others make users specifically invoke network functions.

A **network,** in the simplest terms, is a communication path between two or more systems.

Distributed systems depend on networking for their functionality.

Networks vary by the protocols used, the distances between nodes, and the transport media.

TCP/IP is the most common network protocol, although ATM and other protocols are in widespread use. Likewise, operating system support of protocols varies.

Some operating systems have taken the concept of networks and distributed systems further than the notion of providing network connectivity.

A **network operating system** is an operating system that provides features such as file sharing across the network and that includes a communication schemethat allows different processes on different computers to exchange messages.

1.6 Special-Purpose Systems

The discussion thus far has focused on general-purpose computer systems that we are all familiar with.

There are, however, different classes of computer systems whose functions are more limited and whose objective is to deal with limited computation domains.

1. Real-Time Embedded Systems.

2. Multimedia Systems.

1.7 Computing Environments

So far, we have provided an overview of computer-system organization and major operating-system components.

We conclude with a brief overview of how these are used in a variety of computing environments.

1- Traditional Computing.

2- Client-Server Computing.

3- Peer-to-Peer Computing.

4- Web-Based Computing.


THANKS